A

Appendix A
Building the MSL Driver

Development Process

This appendix describes the process of creating, assembling, linking, and loading a NetWare SFT III Mirrored Server Link driver.

Creating the Source Files

Most MSL drivers are written in 386 assembly code using 32-bit register operations. Chapters 2 through 5 provide the specifications for writing MSL drivers. In addition, Appendix E contains a listing of an MSL driver template that can be used as a base for your driver development. The MSL include file is also required and is listed in Appendix D.

Assembling/Compiling the Source Files

NetWare MSL driver developers must use assemblers/compilers that produce native 32-bit code and object modules compatible with Phar Lap's Easy OMF-386 format. The NetWare Linker (NLMLINK) must be used to link the modules and requires this object module format.

Novell LAN driver developers currently use the 386ASM (v2.0 or later) protected mode assembler by Phar Lap Software, Inc.

Note: Drivers must be assembled with the case sensitive option.

```
386ASMP  <driver>  -fullwarn  -twocase
```

Linking the Object Files

The NetWare linker converts the object file(s) that make up the driver into a super object file which is the NetWare Loadable Module. NLMLINK requires a linker definition file to create an NLM. The linker definition file is described below. To use the linker, type:

```
nlmlink filename
```

(where *filename* is the name of the linker definition file)

Linker Definition File

Each driver module must have a corresponding linker definition file with a .DEF extension. This file is needed by the NetWare linker, NLMLINK. The file contains information about the module including a list of all NetWare variables and routines the driver must access. The definition file also provides directions to NLMLINK for producing the super-object module which is loaded as an NLM by the NetWare OS.

In the example below, NLMLINK will produce a super-object output file (with a .MSL extension) that is the NLM form of the MSL driver, ready to load into an active NetWare environment. The sample definition file directs NLMLINK to find *sample.obj*, link it, and produce the module *sample.msl*.

Additional OS calls may be required in the import list for your MSL driver; these calls are only the ones used in the sample drivers provided by Novell. More OS calls that may be used and imported here are documented in Chapter 5 of this document.

Example

```
type 5                                  /* specify NLM type = MSL driver */
version  1,00
description "NetWare 386 Sample MSL Driver version 1.00 (910822)"
output   sample
input    sample
start    DriverInitialize               /* public routine in NLM      */
check    DriverCheck                    /* check before allowing exit */
exit     DriverRemove                   /* public routine in NLM      */
map                                     /* produces map               */
import                                  /* externals                  */


         AllocateResourceTag
         CancelInterruptTimeCallBack
         CancelNoSleepAESProcessEvent
         CancelSleepAESProcessEvent
         ClearHardwareInterrupt
         CRescheduleLast
         DeRegisterHardwareOptions
         DeRegisterServerCommDriver
         GetCurrentTime
         OutputToScreen
         ParseDriverParameters
         RegisterHardwareOptions
         RegisterServerCommDriver
         ScheduleInterruptTimeCallBack
         ScheduleNoSleepAESProcessEvent
         ScheduleSleepAESProcessEvent
         ServerCommDriverError
         SetHardwareInterrupt

         GetNextPacketPointer
         MaximumCommDriverDataLength
         PacketSizeDriverCanNowHandle
         PacketSizeNowAvailable
         ReceiveServerCommPointer
         SendServerCommCompletedPointer
         ServerCommACKTimeOut
```

Linker Keywords and Parameters

The following keywords are for use in the linker definition file to direct NLMLINK in creating MSL drivers. The keywords are not case sensitive, and can occur in any order. Those keywords that are *required* and those that are *optional* are indicated.

type - (required)
Specifies the type of loadable module as indicated below, and implicitly determines the extension to append on the output file. MSL drivers should use type 5.

1 Lan Driver (.LAN)
2 Disk driver (.DSK)
3 Name space module (.NAM)
4 Utility (.NLM)
5 MSL driver (.MSL)

Note: You will need the NLMLINK executable dated 1-07-92 or later which supports the new type 5 for MSL drivers. The NetWare loader has been modified to load a .MSL file correctly.

description - (required)
Specifies a string that describes the loadable module to be created. The console command MODULES will output this description string. The description can be 1-127 bytes long, must be enclosed in double quotes and may not include a null, double quote, carriage return, or newline. The description should contain the indicated fields in the following order:

```
"name, description, version M.mm, (yymmdd)"
```

Name refers to the company or product name. The *v* of *version* must be in lower case. *M* indicates a major version and *mm* indicates minor version (*all* digits must be present). The *yymmdd* is the year, month, and day. For example, July 17 1991 would be represented as `(910717)`.

output - (required)
Specifies the name of the *output* file (the extension will be added by the linker as specified by the *type* keyword).

input - (required)
Specifies the name of the input .OBJ file(s).

start - (required)
Specifies the name of the loadable module's *initialization* procedure. When the supervisor uses the LOAD console command to load the module, NetWare calls this procedure.

exit - (required)
Specifies the name of the loadable module's *exit* procedure. This procedure is called when the supervisor uses the UNLOAD console command.

check - (optional)
Specifies the name of the loadable module's *check* procedure. The console command UNLOAD calls an NLM's check procedure (if it exists) before unloading the module or downing the server. The check procedure is *optional* for MSL drivers; it provides an opportunity to warn the user what may happen if the *unload* is continued.

import - (required)
Specifies a list of variables and procedure names that are external to the object files. These variables and procedures are NetWare Operating System variables and procedures (or variables and procedures from other loadable modules *which have been previously loaded*) which will be linked to the module after it has been loaded, and before it begins initialization.

export - (optional)
Specifies a list of variables and procedure names resident in the loadable module to be made available to other loadable modules.

map - (optional)
Directs the linker to create a map file.

debug - (optional)
Specifies that the linker will include debug information in the output file. (Be sure to remove this keyword when you are ready to build your final production driver.)

module - (optional)
Specifies loadable modules that must be loaded *before* the current loadable module is loaded. The loader will attempt to find and load any modules not already in the server memory. If it cannot, the current module may not be loaded.

@ operator - (optional)
The @ operator can be use with the *input*, *import*, and *export* directives. It indicates that the list is to be read from a file. This list can be nested; i.e., the file may include another @ operator. The file specifier, including path, must immediately follow the @ operator.

```
import  @file.txt
```

custom - (optional)
Specifies the name of a custom data file that is to be appended to the output file. (An MSL driver may need to use this feature if the MSL adapter contains special customized firmware, for example.)

copyright - (optional)
Tells the linker to include a copyright string in the output file. An ASCII string 1 to 252 bytes long, in double quotes, following the keyword *copyright* is displayed whenever the MSL driver is loaded. To start a new line within the displayed string, use "\n". If the copyright keyword is used but no string is entered, the linker includes the Novell default copyright message.

Note: To use the copyright keyword, you must use the NLMLINKP.EXE.

version - (required)
Specifies the version of the module that should be placed into the header field. The format for this keyword is:

```
version MajorVersion, MinorVersion[, Rev]
```

The version must be separated by commas: *MajorVersion* is 1 digit, *MinorVersion* is two digits. The last comma and the *Rev* are optional. *Rev* is a number from 1-26 representing a letter a-z. For example, `VERSION 3,10,1` in the .DEF file produces the version field *3.10a* in the header of the output file.

Loading and Unloading Drivers

To load a driver, it must first be placed on a floppy diskette, in a directory on the DOS partition of the file server's hard disk, or in the SYS:SYSTEM directory of the file server (the default). The console command *load* is used to load the desired driver into file server memory. The NetWare Loader resolves the driver's import list, loads NLMs specified in the module list, and dynamically links the driver to the NetWare operating system. The following examples show how to load a sample driver from different sources.

Loading from a floppy disk drive:

```
load a:sample.msl port=200 int=5
```

Loading from a local hard disk (note that the file extension is not required in the load command, and that DOS paths may be specified):

```
load c:\NWSFTIII\sample
```

Loading from the file server (only if the SYS volume is mounted):

```
load sample
```

To unload a driver, use the *unload* command as follows:

```
unload sample
```

Unloading a driver causes NetWare to call the driver's remove procedure. The driver is removed from memory after it returns. A reentrant driver must return all resources for all its adapters successfully loaded. (Note: Reentrancy is currently not supported for MSL drivers.) A single call to the *DriverRemove* procedure indicates an *unload* for all associated adapters.

Load Keywords and Parameters

This section describes the parameters used by MSL drivers for the NetWare **LOAD** command. The *ParseDriverParameters* routine handles the load command parameters. The load parameters and examples of their use are described below.

PORT - This is the I/O mapped address base that the user wants the board to use. A port length can also be included as shown in the following examples.

```
LOAD <driver>  PORT=300
LOAD <driver>  PORT=300:A
LOAD <driver>  PORT=300:A  PORT1=700:8
```

MEM - This is the beginning address of the shared RAM that the board can use. The size of the shared memory buffer can also be specified.

```
LOAD <driver>  MEM=C0000
LOAD <driver>  MEM=C0000:1000
LOAD <driver>  MEM=C0000:1000  MEM1=CC000
```

INT - This is the interrupt number that the adapter will use.

```
LOAD <driver>  INT=3
LOAD <driver>  INT=3  INT1=5
```

DMA - If the board supports DMA, this is the Direct Memory Address channel that the NIC should use for data transfer to memory.

```
LOAD <driver>  DMA=0
LOAD <driver>  DMA=0  DMA1=3
```

SLOT - This is a Micro Channel or EISA slot number.

```
LOAD <driver>  SLOT=4
```

RETRIES - This is the number of send retries that the driver should use in its attempts to send packets.

```
RETRIES = n     or     RETR = n
```